

101642,731

CF017487  
US  
hdc

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application: 2 0 0 2 年 8 月 2 1 日

出 願 番 号  
Application Number: 特 願 2 0 0 2 - 2 4 0 5 5 5

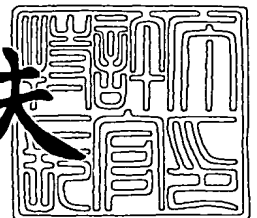
[ST. 10/C]: [ J P 2 0 0 2 - 2 4 0 5 5 5 ]

出 願 人  
Applicant(s): キヤノン株式会社

2 0 0 3 年 9 月 8 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 3 - 3 0 7 3 2 7 1

【書類名】 特許願

【整理番号】 4784002

【提出日】 平成14年 8月21日

【あて先】 特許庁長官 太田 信一郎 殿

【国際特許分類】 G06F 13/00

【発明の名称】 情報処理装置、情報処理方法、制御プログラム

【請求項の数】 9

【発明者】

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会社  
内

【氏名】 江端 正道

【特許出願人】

【識別番号】 000001007

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号

【氏名又は名称】 キャノン株式会社

【代表者】 御手洗 富士夫

【電話番号】 03-3758-2111

【代理人】

【識別番号】 100090538

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会社  
内

【弁理士】

【氏名又は名称】 西山 恵三

【電話番号】 03-3758-2111

**【選任した代理人】****【識別番号】** 100096965**【住所又は居所】** 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会  
社内**【弁理士】****【氏名又は名称】** 内尾 裕一**【電話番号】** 03-3758-2111**【手数料の表示】****【予納台帳番号】** 011224**【納付金額】** 21,000円**【提出物件の目録】****【物件名】** 明細書 1**【物件名】** 図面 1**【物件名】** 要約書 1**【包括委任状番号】** 9908388**【プルーフの要否】** 要

【書類名】 明細書

【発明の名称】 情報処理装置、情報処理方法、制御プログラム

【特許請求の範囲】

【請求項 1】 周辺装置に接続可能な情報処理装置にデバイスドライバを管理する管理方法であって、

情報処理装置内のデバイスドライバのバージョン情報とデバイスドライバが制御可能な周辺装置の情報を取得する取得ステップと、

前記取得ステップにて取得したデバイスドライバのバージョン情報とデバイスドライバが制御可能な周辺装置の情報をを用いて、特定の周辺装置に対応し、複数のバージョンが存在し得るデバイスドライバであって、情報処理装置の記憶装置に存在するデバイスドライバのうち不要となったデバイスドライバを認識する認識ステップと、

前記認識手段によって不要と認識されたデバイスドライバの削除を制御する削除制御ステップを有することを特徴とする管理方法。

【請求項 2】 前記認識ステップは、周辺装置を特定する特定手順と、前記特定手順により特定された周辺装置を制御可能なデバイスドライバのうち、最もバージョンが新しいデバイスドライバを不要ではないと判断する判断手順とを備え、

前記特定手順により特定された周辺装置に対応するデバイスドライバに対して繰り返し前記判断手順における処理を行うことを特徴とする請求項 1 に記載の管理方法。

【請求項 3】 前記認識ステップでは、周辺装置とデバイスドライバの情報を含むテーブルデータとして管理し、該テーブルデータより、不要となったデバイスドライバを求めることを特徴とする請求項 2 に記載の管理方法。

【請求項 4】 新規にデバイスドライバを情報処理装置に追加した後に、前記取得ステップと、前記認識ステップと、前記削除ステップを、実行することを特徴とする請求項 1 乃至 3 のいずれかに記載の管理方法。

【請求項 5】 周辺装置に接続可能な情報処理装置にデバイスドライバを管理可能な情報処理装置であって、

情報処理装置内のデバイスドライバのバージョン情報とデバイスドライバが制御可能な周辺装置の情報を取得する取得手段と、

前記取得手段が取得したデバイスドライバのバージョン情報とデバイスドライバが制御可能な周辺装置の情報をを用いて、特定の周辺装置に対応し、複数のバージョンが存在し得るデバイスドライバであって、情報処理装置の記憶装置に存在するデバイスドライバのうち不要となったデバイスドライバを認識する認識手段と、

前記認識手段によって不要と認識されたデバイスドライバの削除を制御する削除制御手段とを有することを特徴とする情報処理装置。

【請求項 6】 前記認識手段は、特定の周辺装置を制御可能なデバイスドライバのうちで最もバージョンが新しいデバイスドライバを必要ではないと認識することを特徴とする請求項 5 に記載の情報処理装置。

【請求項 7】 前記認識ステップでは、周辺装置とデバイスドライバの情報を含むテーブルデータとして管理し、該テーブルデータより、不要となったデバイスドライバを求めることを特徴とする請求項 6 に記載の情報処理装置。

【請求項 8】 新規にデバイスドライバを情報処理装置に追加した後に、前記取得手段と、前記認識手段と、前記削除手段とを、機能させることを特徴とする請求項 6 乃至 7 のいずれかに記載の情報処理装置。

【請求項 9】 周辺装置に接続可能な情報処理装置にデバイスドライバを管理可能な情報処理装置において実行される制御プログラムであって、

情報処理装置内のデバイスドライバのバージョン情報とデバイスドライバが制御可能な周辺装置の情報を取得する取得ステップと、

前記取得手段が取得したデバイスドライバのバージョン情報とデバイスドライバが制御可能な周辺装置の情報をを用いて、特定の周辺装置に対応し、複数のバージョンが存在し得るデバイスドライバであって、情報処理装置の記憶装置に存在するデバイスドライバのうち不要となったデバイスドライバを認識する認識ステップと、

前記認識ステップによって不要と認識されたデバイスドライバを削除する処理を制御する削除制御ステップとを情報処理装置において実行させることを特徴と

する制御プログラム。

**【発明の詳細な説明】**

**【 0 0 0 1 】**

**【発明の属する技術分野】**

本発明は、周辺装置、周辺装置と接続可能な情報処理装置、及び、これらを含むシステムに関するものである。

**【 0 0 0 2 】**

**【従来の技術】**

一般に、ホストコンピュータと、周辺装置の一例であるプリンタによって構成される印刷システムにおいて印刷を行なう場合、まず、ホストコンピュータ側でアプリケーションを実行して文字や図形などの印刷対象の描画データを作成する。アプリケーションは、メニューなどで印刷処理が選択されたとき、オペレーティングシステムに対して、作成された描画データを渡して印刷処理を依頼する。アプリケーションより印刷処理の依頼があったとき、オペレーティングシステムはアプリケーションより渡された描画データを、ホストコンピュータに接続されたプリンタが実行可能な印刷コマンドへ変換を行なうため、そのプリンタに対応したデバイスドライバの一例であるプリンタドライバを実行する。プリンタドライバは、オペレーティングシステムによって規定された手続きにより、描画データをオペレーティングシステムより受け取り、それを印刷コマンドへ変換する。そしてプリンタドライバはオペレーティングシステムの通信機能を使用して接続されたプリンタへ印刷コマンドの出力を行なう。

**【 0 0 0 3 】**

プリンタは、機種ごとに、サポートしている用紙サイズ、インクの種類、用紙への記録方法、搬送可能な用紙の種類、用紙の搬送方法などの機能に違いがあり、それにより印刷制御方法、画像処理方法、使用できるプリンタコマンドのパラメータが異なる。そのため、アプリケーションの描画データをプリンタの制御コマンドへ変換するプリンタドライバは、プリンタの機種別の機能特性を十分考慮したうえで設計、作成を行わなければならない。よって、プリンタの機能特性の違いをプリンタドライバの制御アルゴリズムに反映させる必要あり、プリンタが

サポートしているコマンド体系が同じであったとしても、制御アルゴリズムを実行するプログラムの違いからプリンタドライバの実行モジュールを各プリンタ個別に作成して供給することが行われている。

#### 【0 0 0 4】

一方、プリンタの機能特性をデータベース化して、機種別に対応しなければならない処理の部分をプリンタドライバはそのデータベースを参照して得た情報に応じて内部アルゴリズムを切り替えることというプログラム方法がある。この場合、一つのプリンタドライバの実行モジュールで複数種類のプリンタをサポートすることが可能である。このようにして、機能特性の異なる複数種類のプリンタに対し、ひとつのプリンタドライバを製品として供給することも行なわれている。

#### 【0 0 0 5】

上述したように、プリンタとそのプリンタの制御を行うプリンタドライバは対応関係があり、使用しているプリンタに適したプリンタドライバをホストコンピュータにインストールする必要がある。プリンタドライバのインストールやアンインストールの作業は、インストーラと呼ばれるプログラムによって行っている。

#### 【0 0 0 6】

##### 【発明が解決しようとする課題】

すでにプリンタドライバがインストールされているホストコンピュータにプリンタドライバをインストールするとき、プリンタドライバのファイル名に変更がない場合は、ファイルの重ね書きを行えばよく、問題は生じない。しかしながら、例えば、プリンタドライバが一つのプリンタのみをサポートするのか、複数のプリンタ機種をサポートするのかについてプリンタドライバの仕様の変更があった場合など、プリンタドライバのファイル名を変更して対応しなければならない場合がある。プリンタドライバのファイル名に変更がある場合は、以前よりホストコンピュータにあったプリンタドライバを削除する必要がある。

#### 【0 0 0 7】

この際、ドライバをアンインストール（削除）するための機能は、プリンタドライバのインストーラの一機能として提供するか、或いは、専用のアンインスト

ールプログラムとして提供することが考えられ、ユーザは、それを使用して手動で煩雑なアンインストール作業を行なうことが必要となる。

#### 【0008】

しかしながら、例えば、一機種をサポートしているタイプのプリンタドライバから、複数機種をサポートするタイプのプリンタドライバへ変更することが必要なシステムの移行作業が行われる場合がある。この場合には、複数機種のプリンタをホストコンピュータに接続して使用しているときは、現在接続されているプリンタが新しい複数機種サポート型のプリンタドライバによりサポートされているかどうかを正確にユーザが逐一把握した上で、既存の不要な機種用のプリンタドライバを選択的に削除するという作業を手動で行わなければならなかった。あるいは、一度すべてのプリンタドライバを削除してから、プリンタドライバをインストール（導入）し直すなどの面倒な作業が必要であった。

#### 【0009】

本発明は上記課題を解決するためになされたものであり、不要となったプリンタドライバを自動的に認識して削除することにより、プリンタドライバのインストール作業を簡便化することができるインストーラの提供を目的とする。

#### 【0010】

##### 【課題を解決するための手段】

上記目的の少なくとも一つを達成するため、本発明は、例えば以下の構成を備える。

#### 【0011】

請求項1に記載の発明は、周辺装置に接続可能な情報処理装置にデバイスドライバを管理する管理方法であって、情報処理装置内のデバイスドライバのバージョン情報とデバイスドライバが制御可能な周辺装置の情報を取得する取得ステップと、前記取得手段が取得したデバイスドライバのバージョン情報とデバイスドライバが制御可能な周辺装置の情報から、特定の周辺装置に対応し、複数のバージョンが存在し得るデバイスドライバのうち、情報処理装置の記憶装置に存在するデバイスドライバのうち、不要となったデバイスドライバを認識する認識ステップと、前記認識手段によって不要と認識されたデバイスドライバの削除を制御



する削除制御ステップを有することを特徴とする。

【0012】

請求項2に記載の発明は、前記認識ステップは、周辺装置を特定する特定手順と、前記特定手順により特定された周辺装置を制御可能なデバイスドライバのうち、最もバージョンが新しいデバイスドライバを不要ではないと判断する判断手順とを備え、前記特定手順により特定された周辺装置に対応するデバイスドライバに対して繰り返し前記判断手順における処理を行うことを特徴とする。

【0013】

請求項3に記載の発明は、前記認識ステップでは、周辺装置とデバイスドライバの情報を含むテーブルデータとして管理し、該テーブルデータより、不要となったデバイスドライバを求めることを特徴とする。

【0014】

請求項4に記載の発明は、新規にデバイスドライバを情報処理装置に追加した後に、前記取得ステップと、前記認識ステップと、前記削除ステップを、実行することを特徴とする。

【0015】

請求項5乃至請求項9に記載の発明は、請求項1乃至4に記載の発明の装置又はプログラムの発明であるので省略する。

【0016】

【発明の実施の形態】

以下、図面を参照して本発明の実施形態を詳細に説明する。

【0017】

図1はプリンタドライバのインストール先であるコンピュータとプリンタとからなるプリントシステムを示すブロック図である。図1において、1はホストコンピュータであり、イーサネット(R)やUSBなどの外部バス10(通信バス)を介してプリンタA11、プリンタB12が接続されている。図1ではプリンタは2つのみ図示したが、通信バス10を介して任意の数のプリンタが接続可能である。

【0018】

プリンタ A 1 1 およびプリンタ B 1 2 は、それぞれプリンタコントローラ 1 3 およびプリンタエンジン 1 4 を備えるインクジェットプリンタであり、種々のサイズのプリント媒体に画像、文字、グラフィクス等をプリント出力することができる。ここで、プリンタコントローラ 1 3 はプリンタ全体の制御を実行する制御回路およびプログラムを意味し、一方、プリンタエンジン 1 4 はインクジェットプリンタヘッドの走査やプリント媒体の搬送を行う機構そのものを意味するものである。プリンタ 1 1 およびプリンタ 1 2 は、それぞれインクジェットカートリッジを備え、インクジェットカートリッジはインクジェットヘッドのヘッドに供給するインクを貯留したインクタンクを色毎に組合せて一体化したものである。

#### 【0019】

ホストコンピュータ 1 において、2 は CPU であり、起動時 ROM 3 に格納されたプログラムによりハードディスクや CD-ROM などの外部記憶装置 6 に格納されたオペレーティングシステム（以下、図面では OS とすることがある。）を RAM 4 にロードし実行してシステムを制御する。5 は CRT あるいは LCD などの表示装置であり、CPU 1 の制御にしたがって各種の表示を行うことができる。7 はキーボード、マウスなどの入力装置であり、CPU 1 への各種入力を行う。9 はホストコンピュータ内部バスを表し、CPU 2、ROM 3、RAM 4、表示装置 5、外部記憶装置 6、入力装置 7、外部 I/F 8 どの間でデータのやり取りを可能にするものである。

#### 【0020】

外部記憶装置 6 には、オペレーティングシステムの他にアプリケーションプログラムやプリンタドライバなどの各種プログラムが格納されており、CPU 1 は種々のアプリケーションプログラムを実行し、オペレーティングシステムの制御に従い、プリンタ A 1 1、プリンタ B 1 2 に対応したプリンタドライバを実行して、アプリケーションプログラムの描画データをプリンタ A 1 1、プリンタ B 1 2 が実行可能なプリンタ制御コマンドに変換する。プリンタドライバの制御にもとづき CPU 1 が生成したプリンタ制御コマンドは、I/F 8 により、通信バス 1 0 に出力され、プリンタ A 1 1 またはプリンタ B 1 2 へ送信される。プリンタ A 1 1 またはプリンタ B 1 2 はこのようにしてホストコンピュータ 1 から送信さ

れたプリンタ制御コマンドの受信を行い、印刷を行う。

#### 【0021】

図2は、コンピュータのオペレーティングシステムにおける印刷処理に関するプログラム実行モジュールの内部ブロック図を表す。アプリケーション15は、あらかじめユーザの操作により作成された描画データを、オペレーティングシステム16に渡すとともにオペレーティングシステム16に対し印刷処理の依頼を行う。オペレーティングシステム16はアプリケーションから渡された描画データを印刷スプールファイル17にセーブする。アプリケーションから描画データをすべて受け取り印刷スプールファイル17に保存した後、オペレーティングシステム16は出力先のプリンタに対応したプリンタドライバ18をロードする。オペレーティングシステム16は印刷スプールファイル17から描画データをロードし、プリンタドライバ18に渡し処理を依頼する。プリンタドライバ18は渡された描画データをプリンタコマンドへ変換して、オペレーティングシステム16の通信機能を利用してプリンタ19へ出力する。プリンタ19ではこのようにして生成された印刷コマンドにもとづいて印刷を行う。

#### 【0022】

図3は、本実施形態のインストーラが実行されるときの実行環境例のブロック図を示す。図3はプリンタドライバ1、プリンタドライバ2がすでにホストコンピュータ上にインストールされていた状態で、新たにプリンタドライバ3がインストールされた状態を示している。インストーラ20は、インストール作業の実行を担当するインストールメインモジュール21と、プリンタドライバのバージョンとサポートプリンタの情報を取得して処理を行うバージョン管理モジュール22で構成される。プリンタドライバ1は、バージョンが1.0.0の制御プログラムによるプリンタドライバであり、プリンタAをサポートする。プリンタドライバ2は、バージョン1.1.0の制御プログラムを持つプリンタドライバであり、プリンタDをサポートする。プリンタドライバ3は、バージョン2.0.0の制御プログラムを持つプリンタドライバであり、サポート情報テーブルにセーブされたサポート機種情報のプリンタをすべてサポートすることが可能なプリンタドライバである。

**【0023】**

図5は、サポート情報テーブルのデータ構造を表す。サポート情報テーブルは、プリンタドライバがサポートしているプリンタの機種数と機種の情報を持つ。図5の表のプリンタモジュール3の欄が、プリンタドライバ3に関するものであり、サポート可能なプリンタのデータが保存されている。本実施例では、プリンタドライバ3は、プリンタA、プリンタB、プリンタCの3機種を制御可能である。

**【0024】**

本実施形態のインストーラが実行されると、まず、インストールメインモジュール21がホストコンピュータのCPUにより実行される。図6は、インストールメインモジュール21の処理を表すフローチャートである。ステップS1において、プリンタドライバをオペレーティングシステムで規定された所定の場所にインストールする。ステップS2において、バージョン管理モジュール22を呼び出し、各プリンタドライバのバージョンとサポートしているプリンタ機種情報を取得して、プリンタドライバ管理テーブルBを作成させ取得する。

**【0025】**

図7は、ステップS2の詳細であり、バージョン管理モジュール22の処理を表すフローチャートである。ステップS9において、ホストコンピュータの外部記憶装置上に存在するプリンタドライバの検索を行なう。この処理は、例えば、プリンタドライバをオペレーティングシステムに規定された場所の範囲内で、あらかじめ決めておいたディレクトリの中にインストールしておくものとし、バージョン管理モジュール22がそのディレクトリの中の実行ファイルを検索することにより行われる。ステップS10において、検索で見つかったプリンタドライバの総数を変数nに格納する。ステップS11で、カウンタ用の変数iを0に初期化する。ステップS12で、カウンタ変数iとプリンタドライバの総数nを比較する。i < nのときは、ステップS13に進む。ステップS13において、プリンタドライバのバージョン情報を取得する。例えば、Mac OS X（商標）のようなOSにおいては、プリンタドライバはバンドルと呼ばれるディレクトリ階層構造で構成される。ディレクトリ下の所定の場所に所定のファイル名のフ

ファイルがあり、その中にバージョン番号が記述されている。あるいは、Mac OS 9（商標）のようなオペレーティングシステムでは、リソースと呼ばれるファイル領域にバージョン情報が記述されている。ステップS13では、例えば、アプリケーションとして提供されるインストーラが、これらOSが管理する情報を参照してバージョン情報を取得する。

#### 【0026】

ステップS14では、インストーラは、i番目のプリンタドライバのサポート情報テーブルを検索する。例えば、Mac OS X（Apple社の商標）用ドライバのようなディレクトリ構造で管理されるプリンタドライバは、プリンタドライバのディレクトリ階層下の所定の場所にサポート情報ファイルを格納されるものとする。別のオペレーティングシステムのプリンタドライバでは、リソースと呼ばれるファイル領域にサポート情報を保存しておくこともできる。ステップS14では、そのようなサポート情報テーブルの検索処理を行う。サポート情報テーブルが見つかった場合、ステップS15へ進む。ステップS15において、サポート情報テーブルの中に記述されたそのプリンタドライバが制御可能なプリンタの機種を取得する。S14において、サポート情報テーブルが見つからなかった場合、S16に進む。S16では、他の手段によりプリンタドライバがサポートするプリンタ機種情報を取得する。例えば、プリンタドライバのファイル名にプリンタの機種名が含まれるようなときは、バージョン管理モジュールはファイル名からプリンタドライバがサポートする機種名称を取得することができる。また、ファイル名にプリンタの機種名称が含まれていない場合でも、バージョン管理モジュールが、プリンタドライバのファイル名とプリンタ機種を対応付けるテーブルを持ち、これを参照することでプリンタドライバが制御可能なプリンタの機種を知ることができる。

#### 【0027】

ステップS17において、i番目のプリンタドライバがサポートするプリンタ機種情報とプリンタドライバのバージョン番号により、モジュール管理テーブルAを更新する。

#### 【0028】

図9、図10はモジュール管理テーブルAの構造を表す図である。モジュール管理テーブルAは、プリンタの機種を表す情報、プリンタドライバのバージョン番号、プリンタドライバを識別するための情報を持つ。プリンタドライバを識別するための情報はオペレーティングシステムで管理されるプリンタドライバの参照情報である。

### 【0029】

図8はステップS17の詳細を表すフローチャートである。ステップS20において、現在処理対象となっているプリンタドライバがサポートするプリンタの機種数を変数mに代入する。ステップ21において、カウンタ変数jを0に初期化する。ステップ22において、変数mと変数jの大小関係を比較する。j < mのときは、ステップS23へ進み、プリンタドライバがサポートするプリンタのうち、j番目のプリンタに関してステップS23以降の処理を行う。ステップS23において、j番目のプリンタがプリンタドライバ管理テーブルAに登録済であるかどうか調べる。ステップS23で、j番目のプリンタがプリンタドライバ管理テーブルAに存在しないときは、そのプリンタの情報をモジュール管理テーブルAに登録するため、ステップS25へ進む。j番目のプリンタがプリンタドライバ管理テーブルAに存在するときは、ステップS24へ進む。ステップS24において、対象となっているj番目のプリンタに関して、プリンタドライバ管理テーブルAにすでに登録されているプリンタドライバのバージョンと、現在処理対象となっているプリンタドライバのバージョンを比較する。現在処理中のプリンタドライバのバージョンの方が大きいときは、ステップS25に進む。ステップS25では、プリンタドライバ管理テーブルAの該当プリンタの情報を現在処理対象となっているプリンタドライバの識別情報とバージョン情報で更新する。プリンタドライバ管理テーブルAにすでに登録されているプリンタドライバのバージョンよりも現在処理中のプリンタドライバのバージョンの方が小さいときは、S25のプリンタドライバ管理テーブルAの更新処理は行わない。ステップS26で、カウンタ変数jをインクリメントしてステップS22に戻り、処理対象となっているプリンタドライバの次のサポートプリンタに対して同様の処理を行う。ステップS22において、j < mでなくなり、すべてのプリンタについての

処理が終了したときは、図 8 のフローチャートの処理を終了して、図 7 のフローチャートのステップ S 18 へ戻る。

#### 【0030】

図 7 のフローチャートのステップ S 18 において、カウンタ変数  $i$  をインクリメントしてステップ S 12 へ戻り、次のプリンタドライバに関して同様の処理を行う。ステップ S 12 にて、 $i < n$  でなくなり、すべてのプリンタドライバについて処理が終了したときは、ステップ S 19 へ進む。ステップ S 19 において、プリンタドライバ管理テーブル A からプリンタドライバ管理テーブル B を作成する。

#### 【0031】

図 11、図 12 はプリンタドライバ管理テーブル B の構造を表す。プリンタドライバ管理テーブル B は、プリンタドライバ識別情報、プリンタドライバのバージョン番号、プリンタの機種数、プリンタの情報をもち、プリンタドライバをキーとしてプリンタドライバ管理テーブル A を並びかえたものである。そして、このテーブルを参照すれば、各バージョンのプリンタドライバを必要とするプリンタの機種がわかる。

#### 【0032】

図 3 のプリンタドライバの構成において、バージョン管理モジュール 22 が図 7、図 8 のフローチャートの処理を行い作成されたプリンタドライバ管理テーブル A とプリンタドライバ管理テーブル B が、図 9、図 11 である。図 9 のプリンタドライバ管理テーブル A によれば、プリンタ A をサポートする最もバージョンの大きいプリンタドライバはプリンタドライバ 3 であり、プリンタ B をサポートする最もバージョンの大きいプリンタドライバはプリンタドライバ 3 であり、プリンタ C をサポートする最もバージョンの大きいプリンタドライバはプリンタドライバ 3 であり、プリンタ D をサポートする最もバージョンの大きいプリンタドライバはプリンタドライバ 2 であることがわかる。さらにこのテーブルをもとに作成された図 11 のプリンタドライバ管理テーブル B によれば、プリンタドライバ 1 を必要とするプリンタはなく、プリンタドライバ 2 はプリンタ D のために必要であり、プリンタドライバ 3 はプリンタ A、プリンタ B、プリンタ C のために

必要であることがわかる。

#### 【0033】

バージョン管理モジュールがプリンタドライバ管理テーブルBを作成後、インストールメインモジュール21の処理の図6のステップS3に戻る。ステップS3において、プリンタドライバ管理テーブルBに登録されたプリンタドライバの総数を変数nに代入する。ステップS4において、カウンタ変数iを0に初期化する。ステップS5において、変数nと変数iの大小関係を比較する。i < nのときは、ステップS6へ進む。ステップS6において、プリンタドライバ管理テーブルBのi番目のプリンタドライバを必要とするプリンタの機種数を取得して機種数が0かどうかを調べる。機種数が0のときは、ステップS7へ進み、そのプリンタドライバの削除処理を行う。ステップS6において、i番目のプリンタドライバを必要とするプリンタの機種数が0でないときは、そのプリンタドライバの削除は行わない。

#### 【0034】

ステップS8において、カウンタ変数iをインクリメントして、ステップS5へ戻る。プリンタドライバ管理テーブルBに登録されているすべてのプリンタドライバの処理が終了して、ステップS5において、i < nではなくなったときは、インストールメインモジュール21の処理を終了する。

#### 【0035】

図3のようなプリンタドライバの構成では、プリンタドライバ管理モジュールBは図11のようであるので、プリンタ機種数が0であるプリンタドライバ1は、インストールメインモジュール21の実行後、削除される。

#### 【0036】

図4は、本実施形態のインストーラが実行されるとき、他の実行環境例のブロック図を示す。図4は図3の環境にプリンタドライバ3がインストールされプリンタドライバ1が削除された状態において、さらに、プリンタドライバ4がインストールされたときの状態を表している。プリンタドライバ4は、サポート情報テーブルを持ち、図5のプリンタドライバ4の欄に示すように、プリンタA、プリンタB、プリンタC、プリンタDの4機種を制御可能である。また、プリン



タドライバ4の制御プログラムのバージョンは3.0.0である。

#### 【0037】

このとき、図7、図8のフローチャートのバージョン管理モジュールが実行されることにより作成されるモジュール管理テーブルAは図10である。プリンタA、プリンタB、プリンタC、プリンタDをサポートする最もバージョンの大きいプリンタドライバは、ともにプリンタドライバ4であることがわかる。図10のプリンタドライバ管理テーブルAをもとに作成されるプリンタドライバ管理テーブルBを図12に示す。プリンタドライバ2、プリンタドライバ3を必要とするプリンタは存在せず、プリンタドライバ4を必要とするプリンタは、プリンタA、プリンタB、プリンタC、プリンタDであることがわかる。すなわち、図6のフローチャートにもとづくインストーラメインモジュールを実行すると、プリンタドライバ4がインストールされた後に、プリンタドライバ2とプリンタドライバ3は削除される。

#### 【0038】

(他の実施形態)

本実施形態における各図に示す処理が、外部からインストールされるプログラムによって、各情報処理装置により遂行される。そして、その場合、CD-ROMやフラッシュメモリやFD等の記憶媒体により、あるいはネットワークを介して外部の記憶媒体から、プログラムを含む情報群をホストコンピュータに供給される場合でも本発明は適用されるものである。

#### 【0039】

以上のように、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、又は、外部サーバ（図示省略）からダウンロードすることで、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、本発明の目的が達成されることは言うまでもない。

#### 【0040】

この場合、記憶媒体から読み出されたプログラムコード自体が本発明の新規な

機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。プログラムコードを供給するための記憶媒体としては、たとえば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、DVD、CD-ROM、磁気テープ、不揮発性のメモリカード、ROM、EEPROM等を用いることができる。

#### 【0041】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPU等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

#### 【0042】

以上説明したように、本発明の上述の各実施形態によれば、プリンタドライバのバージョンや制御可能なプリンタなどの違いによりファイル名が異なるプリンタドライバがインストールされているホストコンピュータの環境において、プリンタドライバをインストールした後、不要となったバージョンの古いプリンタドライバを認識して削除することができるプリンタドライバのインストーラを提供できる。これにより、従来ユーザが手動で行っていた不要となった古いバージョンのプリンタドライバを削除する作業をインストーラが行えるようになり、ユーザによるプリンタドライバのアンインストール作業を簡便化することができる。

#### 【0043】

##### 【発明の効果】

不要となったプリンタドライバを自動的に認識し、認識したドライバを削除するので、プリンタドライバのアンインストール作業を簡便化することができる。

**【図面の簡単な説明】****【図 1】**

コンピュータとプリンタとからなる印刷システムの構成の一例を示すブロック図である。

**【図 2】**

印刷システムにおいてオペレーティングシステムを介した印刷処理を表すブロック図である。

**【図 3】**

インストーラが実行される環境の例を表すブロック図である。

**【図 4】**

インストーラが実行される環境の他の例を表すブロック図である。

**【図 5】**

プリンタドライバが制御可能なプリンタの情報を持つサポート情報テーブルを示す図である。

**【図 6】**

インストーラメインモジュールの処理を表すフローチャートである。

**【図 7】**

バージョン管理モジュールの処理を表すフローチャートである。

**【図 8】**

図 7 において、プリンタドライバ管理テーブル A の更新処理の詳細を表すフローチャートである。

**【図 9】**

図 3 のインストーラ実行環境例において作成されるプリンタドライバ管理テーブル A を示す図である。

**【図 1 0】**

図 4 のインストーラ実行環境例において作成されるプリンタドライバ管理テーブル A を示す図である。

**【図 1 1】**

図 3 のインストーラ実行環境例において作成されるプリンタドライバ管理テー

ブル B を示す図である。

【図 1 2】

図 4 のインストーラ実行環境例において作成されるプリンタドライバ管理テーブル B を示す図である。

【符号の説明】

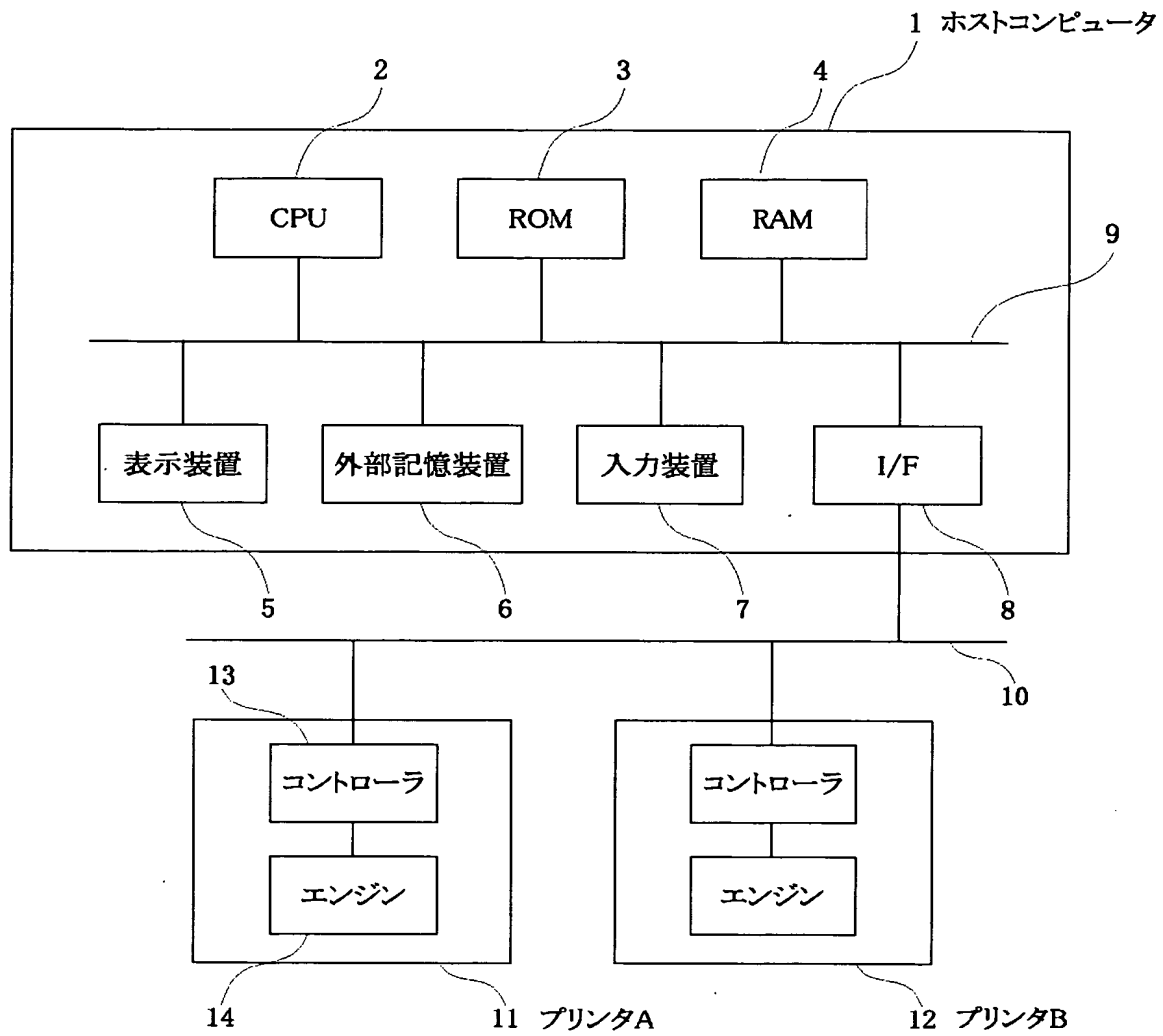
- 1    ホストコンピュータ
- 2    C P U
- 3    R O M
- 4    R A M
- 5    表示装置
- 6    外部記憶装置
- 7    入力装置
- 8    外部 I / F
- 9    内部バス
- 1 0   外部バス
- 1 1   プリンタ A
- 1 2   プリンタ B
- 1 3   プリンタコントローラ
- 1 4   プリンタエンジン
- 1 5   アプリケーション
- 1 6   オペレーティングシステム
- 1 7   印刷スプールファイル
- 1 8   プリンタドライバ
- 1 9   プリンタ
- 2 0   インストーラ
- 2 1   インストールメインモジュール
- 2 2   バージョン管理モジュール
- 2 3   プリンタドライバ 1
- 2 4   プリンタドライバ 2

2 5 プリンタドライバ 3

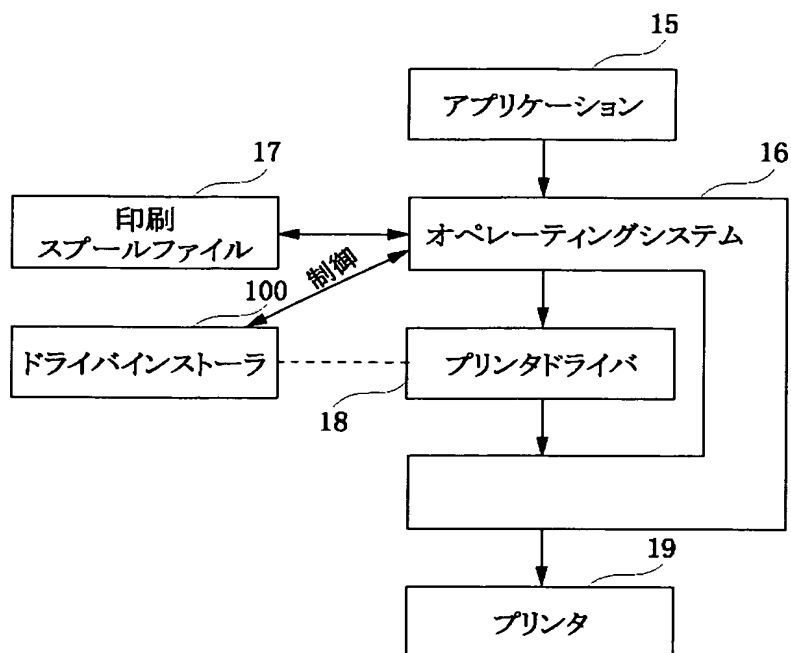
2 6 プリンタドライバ 4

【書類名】 図面

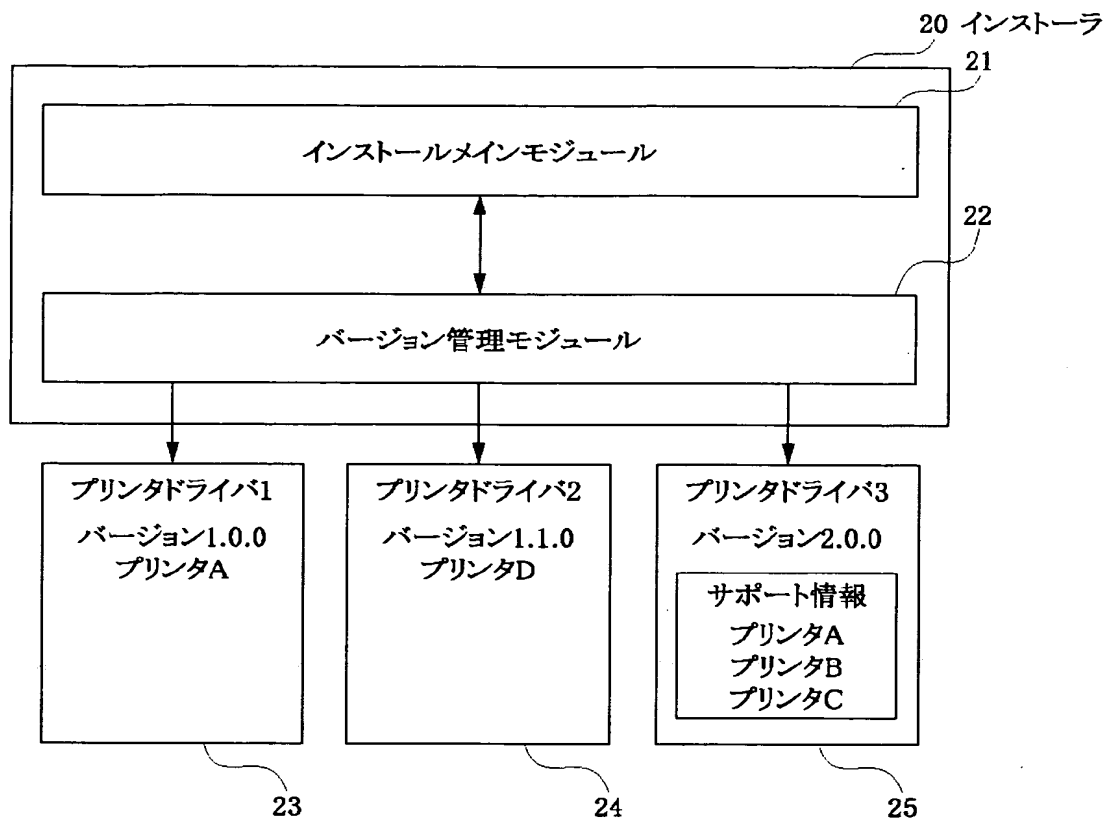
【図 1】



【図 2】

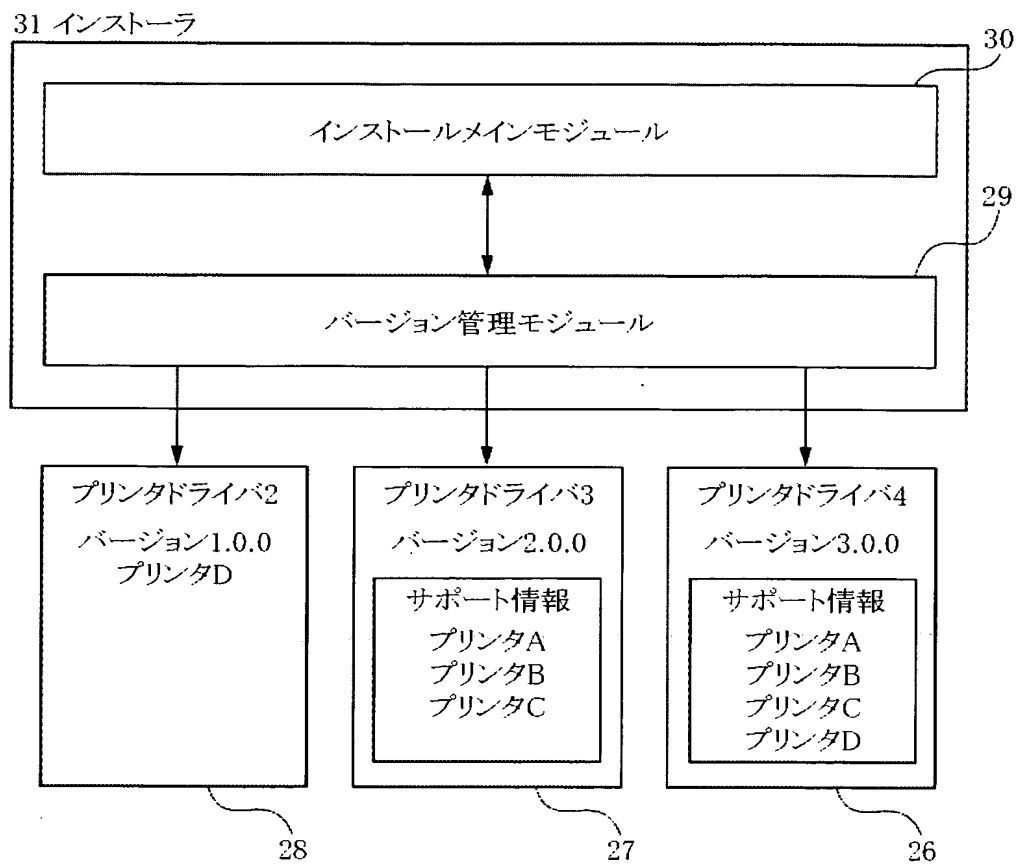


【図 3】





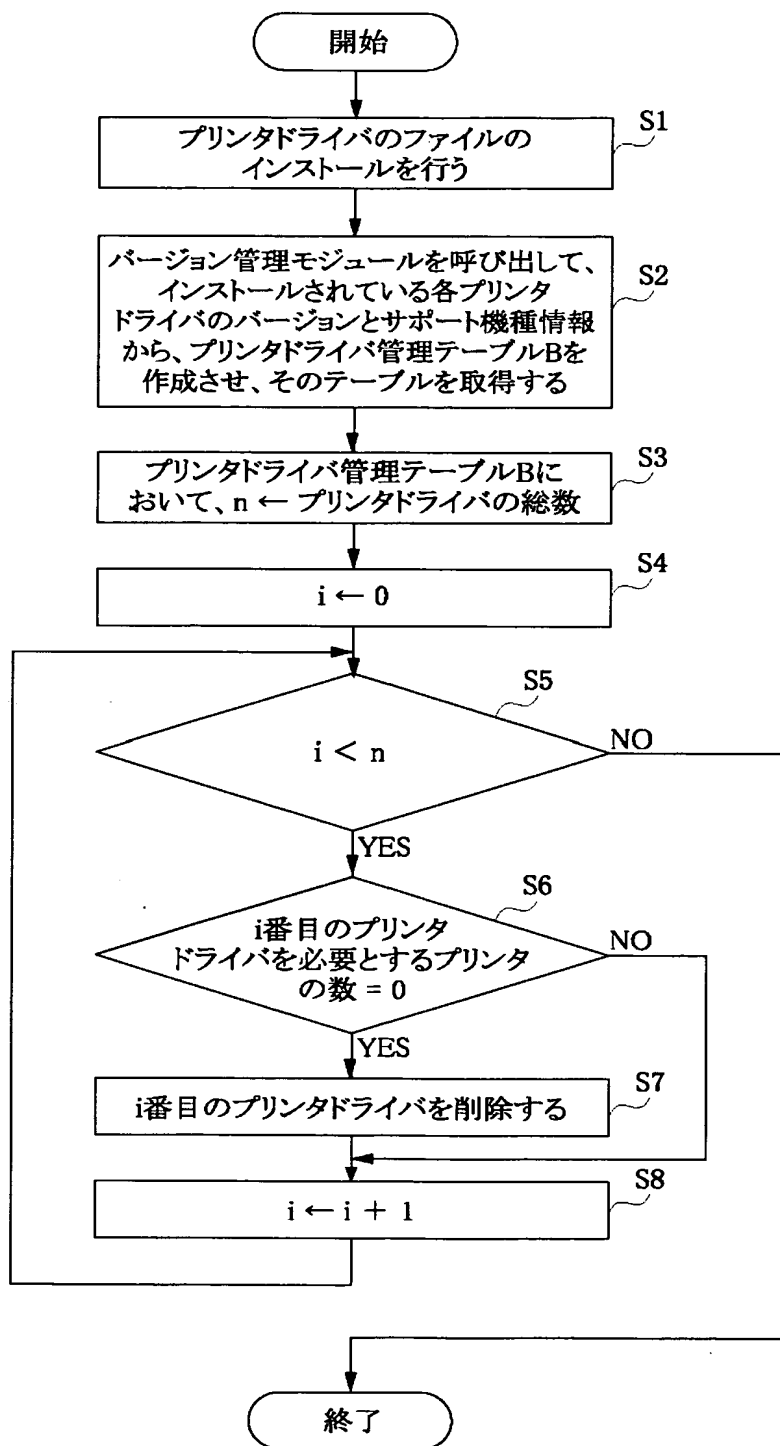
【図 4】



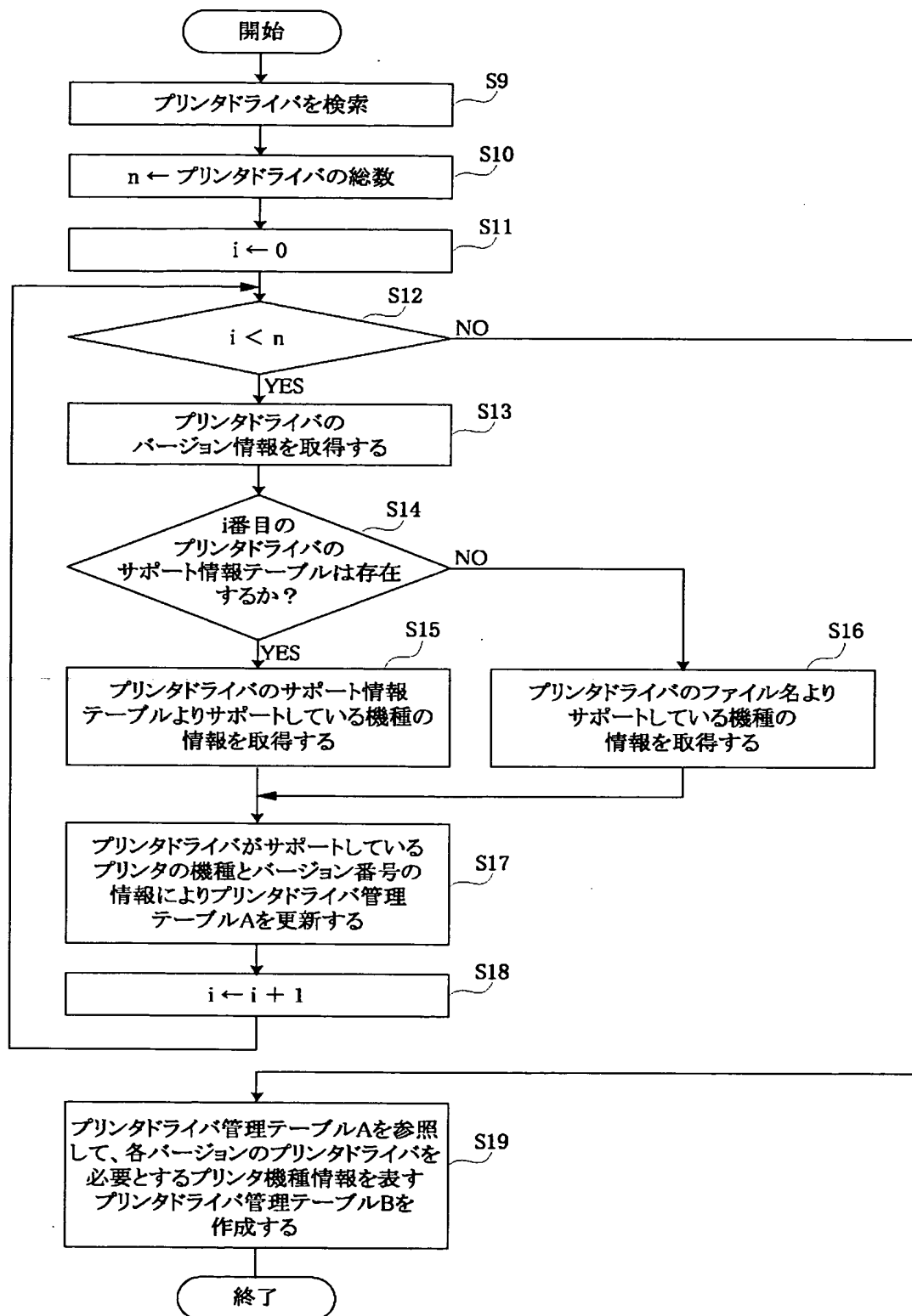
【図 5】

プリンタドライバ	バージョン番号	サポート情報	
		サポート機種数	サポートプリンタ
プリンタドライバ3	2.0.0	3	プリンタA プリンタB プリンタC
プリンタドライバ4	3.0.0	4	プリンタA プリンタB プリンタC プリンタD

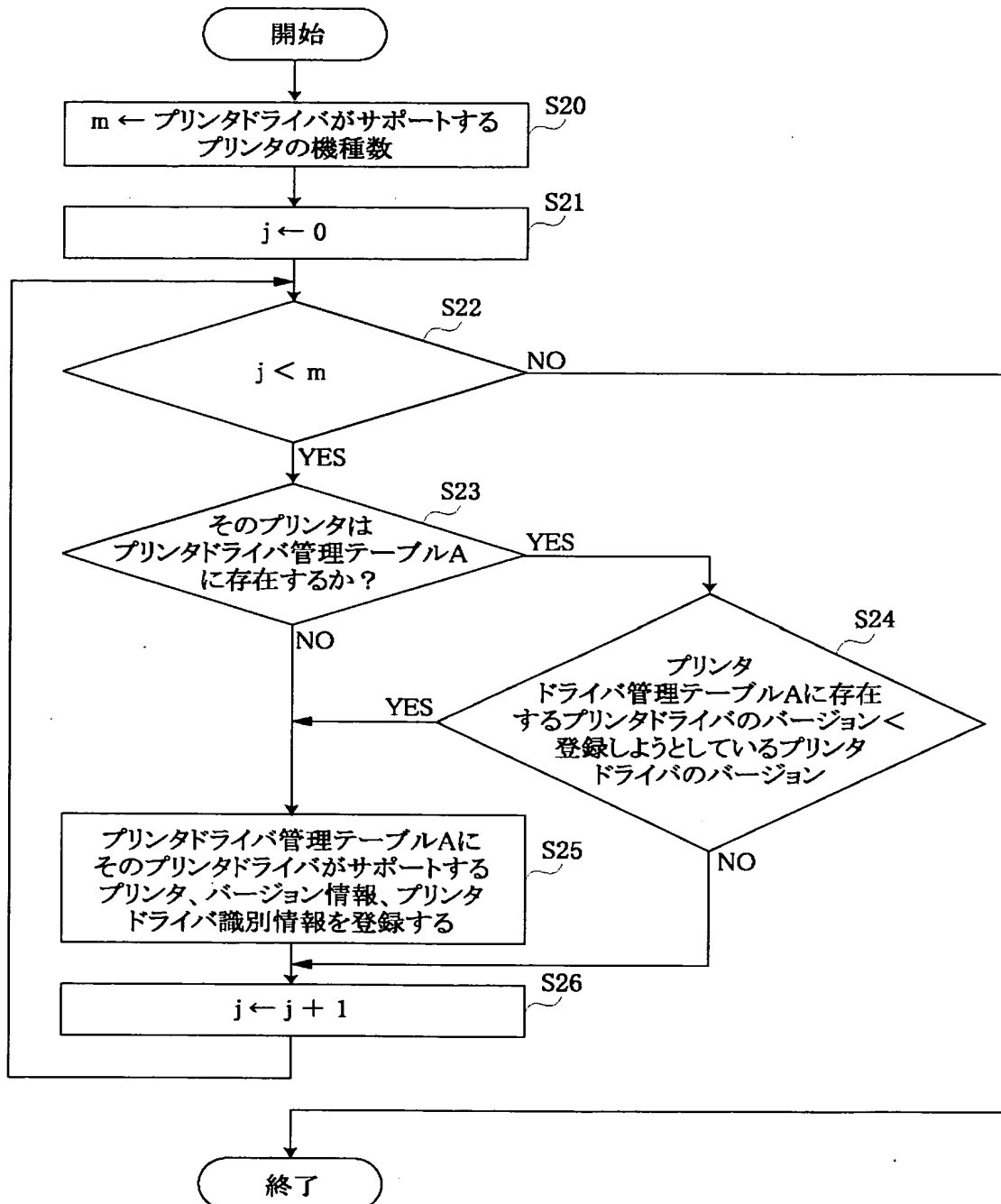
【図 6】



【図 7】



【図 8】



【図 9】

プリンタ	バージョン情報	プリンタドライバ識別情報
プリンタA	2.0.0	プリンタドライバ3
プリンタB	2.0.0	プリンタドライバ3
プリンタC	2.0.0	プリンタドライバ3
プリンタD	1.1.0	プリンタドライバ2

【図 1 0】

プリンタ	バージョン情報	プリンタドライバ識別情報
プリンタA	3.0.0	プリンタドライバ4
プリンタB	3.0.0	プリンタドライバ4
プリンタC	3.0.0	プリンタドライバ4
プリンタD	3.0.0	プリンタドライバ4

【図 1 1】

プリンタドライバ識別情報	バージョン番号	プリンタ機種数	プリンタ
プリンタドライバ1	1.0.0	0	なし
プリンタドライバ2	1.1.0	1	プリンタD
プリンタドライバ3	2.0.0	3	プリンタA プリンタB プリンタC



【図 1 2】

プリンタドライバ識別情報	バージョン番号	プリンタ機種数	プリンタ
プリンタドライバ2	1.1.0	0	なし
プリンタドライバ3	2.0.0	0	なし
プリンタドライバ4	3.0.0	4	プリンタA プリンタB プリンタC プリンタD

【書類名】 要約書

【要約】

【課題】 不要となったプリンタドライバを自動的に認識し、認識したドライバを削除する仕組みを提供し、プリンタドライバのアンインストール作業を簡便化する。

【解決手段】 例えば、ホストコンピュータ 1 0 0 0 内のデバイスドライバのバージョン情報とデバイスドライバが制御可能な周辺装置の情報(図 1 1、図 1 2 のプリンタドライバ管理テーブル B)を取得する取得ステップと、前記取得ステップが取得したデバイスドライバのバージョン情報とデバイスドライバが制御可能な周辺装置の情報を用いて、特定の周辺装置に対応し、複数のバージョンが存在し得るデバイスドライバであって、情報処理装置の記憶装置に存在するデバイスドライバのうち不要となったデバイスドライバを認識する認識ステップと、前記認識手段によって不要と認識されたデバイスドライバの削除を制御する削除制御ステップを有することを特徴とする管理方法を提供する。

【選択図】 図 6

特願 2 0 0 2 - 2 4 0 5 5 5

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 1 0 0 7 ]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都大田区下丸子 3 丁目 3 0 番 2 号

氏 名

キャノン株式会社